AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1   1. (Currently amended) A method for providing object change information
2   from a first system to a second system for synchronizing the second system with
3   the first system, the second system having an object cache for storing objects, the
4   method comprising the steps of:
5   changing an object in the first system;
6   determining an object change set which represents changes made to the
7   object in the first system; and, wherein the object change set includes:
8   a primary key value that identifies the object; and
9   a set of attribute changes which contain the attribute names
10   and the new attribute values of attributes that were changed in the
11   object; and
12   sending the object change set directly from the first system to the second
13   system to cause the second system to apply the object change set to the
14   corresponding object in the second system's cache so as to synchronize the second
15   system with the first system, wherein control is returned to a client coupled to the
16   first system whether or not the second system has completed processing the object
17   change set, wherein the second system is registered in the first system prior to
18   sending the object change information from the first system to the second system,
19   wherein and wherein the registration process causes the first system to know that
20   the second system wants to receive object change information of objects changed

2

21   in the first system, thereby eliminating the need for the second system to register

22   with every object it is interested in.

1   2. (Original) The method as claimed in claim 1 further comprising a step

2   of establishing a communication link between the first system and the second

3   system wherein the distributing step distributes the object change information

4   from the first system to the second system through the communication link.

1   3. (Original) The method as claimed in claim 2 wherein the establishing

2   step establishes the communication link based on a publish/subscribe protocol.

1   4. (Canceled).

1   5. (Original) The method as claimed in claim 1 further comprising a step

2   of sending the object change information to a database for updating the object in

3   the database with the object change information.

1   6. (Original) The method as claimed in claim 5 further comprising the

2   steps of:

3   receiving an error message from the database when the updating of the

4   object in the database fails; and

5   discarding the object change information prior to the distributing step in

6   response to the error message.

1   7. (Original) The method as claimed in claim 1 wherein the first system

2   includes an object cache for storing one or more objects, and the method further

3   comprises a step of merging the object change information into the object cache of

4   the first system.

3

1   8. (Canceled).

1   9. (Canceled).

1   10. (Canceled).

1   11. (Original) The method as claimed in claim 1 wherein the first system

2 includes a cache for storing one or more objects, the method further comprising

3 the steps of:

4   receiving object change information distributed from the second

5 system and containing information of changes made to one or more objects in the

6 second system; and

7   merging the object change information received from the second

8 system into the objects in the cache of the first system to synchronize the first

9 system with the second system.

1   12. (Currently amended) A method for providing object change

2 information from a first system to a second system for synchronizing the second

3 system with the first system, the first system having a first object cache for storing

4 one or more objects and the second system having a second object cache for

5 storing one or more_objects, the method comprising the steps of:

6   determining object change sets which represent changes made to objects in

7 the first system; and, wherein an object change set includes:

8   a primary key value that identifies the object; and

9   a set of attribute changes which contains the attribute

10   names and the new attribute values of attributes that were changed

11   in the object; and

4

12   sending the object change sets ~~directly~~ from the first system to the second

13   system to cause the second system to apply the object change sets to

14   corresponding objects in the second object cache so as to synchronize the objects

15   in the second cache of the second system with the changed objects in the first

16   system, <u>wherein control is returned to a client coupled to the first system whether</u>

17   <u>or not the second system has completed processing the object change set,</u> wherein

18   the second system is registered in the first system prior to sending the object

19   change information from the first system to the second system, ~~wherein~~ <u>and where</u>

20   the registration process causes the first system to know that the second system

21   wants to receive object change information of objects changed in the first system,

22   thereby eliminating the need for the second system to register with every object it

23   is interested in.

1   13. (Original) The method as claimed in claim 12 further comprising a

2   step of establishing a communication link between the first system and the second

3   system wherein the distributing step distributes the object change information

4   from the first system to the second system through the communication link.

1   14. (Canceled).

1   15. (Original) The method as claimed in claim 12 further comprising a

2   step of sending the object change information from the first system to a database

3   for updating the object in the database with the object change information.

1   16. (Original) The method as claimed in claim 15 further comprising the

2   steps of:

5

3   receiving an error message from the database when the updating of

4   the object in the database fails; and

5   discarding the object change information prior to the distributing

6   step in response to the error message.

1   17. (Canceled).

1   18. (Canceled).

1   19. (Canceled).

1   20. (Original) The method as claimed in claim 12 further comprising steps

2   of:

3   receiving object change information distributed from the second

4   system and containing information of changes made to one or more objects in the

5   second system; and

6   merging the object change information received from the second

7   system into the objects in the first cache of the first system to synchronize the first

8   system with the second system.

1   21. (Currently amended) A synchronization executor for providing object

2   change information from a first system to a second system for synchronizing the

3   second system with the first system, the first system being capable of changing the

4   object, the second system having an object cache for storing objects, the system

5   comprising;

6   a synchronization manager for obtaining an object change set which

7   represents changes made to an object in the first system; and, wherein the object

8   change set includes:

9   a primary key value that identifies the object; and

10   a set of attribute changes which contain the attribute names

11   and the new attribute values of attributes that were changed in the

12   object; and

13   a dispatcher for distributing the object change set directly from the first

14 system to the second system to cause the second system to apply the object change

15 set to the corresponding object in the second system's cache so as to synchronize

16 the object in the second system with the first system, wherein control is returned

17 to a client coupled to the first system whether or not the second system has

18 completed processing the object change set, wherein the second system is

19 registered in the first system prior to distributing the object change information

20 from the first system to the second system, wherein and wherein the registration

21 process causes the first system to know that the second system wants to receive

22 object change information of objects changed in the first system, thereby

23 eliminating the need for the second system to register with every object it is

24 interested in.

1   22. (Original) The executor as claimed in claim 21 wherein the

2   synchronization manager establishes a communication link with the second

3   system and the dispatcher distributes the object change information to the second

4   system through the established communication link.

7

1   23. (Previously presented) The executor as claimed in claim 22 wherein
2   the synchronization manager establishes the communication link based on a
3   publish/subscribe protocol.

1   24. (Original) The executor as claimed in claim 21 further comprising a
2   connector for obtaining the object change information that is distributed from the
3   second system.

1   25. (Canceled).

1   26. (Canceled).

1   27. (Canceled).

1   28. (Currently amended) A persistence system for synchronizing an object
2   on a network, the network including a caching system having an object cache for
3   storing objects, the persistence system comprising;
4   a transaction manager for changing an object, determining an object
5   change set which represents changes made to the object; and, wherein the object
6   change set includes:
7   a primary key value that identifies the object; and
8   a set of attribute changes which contain the attribute names
9   and the new attribute values of attributes that were changed in the
10   object; and
11   a synchronization executor for obtaining the object change set from the
12 transaction manager and distributing the object change set to the caching system
13 to cause the caching system to apply the object change set to the corresponding

8

14 object in the cache so as to synchronize the object in the object cache with the

15 changed object in the persistence system, <u>wherein control is returned to a client</u>

16 <u>coupled to the first system whether or not the second system has completed</u>

17 <u>processing the object change set,</u> wherein the second system is registered in the

18 first system prior to distributing the object change information directly from the

19 first system to the second system, ~~wherein~~ <u>and wherein</u> the registration process

20 causes the first system to know that the second system wants to receive object

21 change information of objects changed in the first system, thereby eliminating the

22 need for the second system to register with every object it is interested in.

1   29. (Currently amended) The system as claimed in claim 28 further

2 comprising a persistence system cache for storing one or more objects.

1   30. (Canceled).

1   31. (Original) The system as claimed in claim 28 wherein the

2 synchronization executor establishes the network, and the dispatcher distributes

3 the object change information via the established network.

1   32. (Currently amended) Computer readable media storing instructions for

2 use in the execution in a computer of a method for providing object change

3 information from a first system to a second system for synchronizing the second

4 system with the first system, the second system having an object cache for storing

5 objects, the method comprising the steps of:

6   changing an object in the first system;

7   determining an object change set which represents changes made to the

8 object in the first system<u>;</u> ~~wherein the object change set includes:~~

9

9   ~~a primary key value that identifies the object; and~~

10   ~~a set of attribute changes which contain the attribute names~~

11   ~~and the new attribute values of attributes that were changed in the~~

12   ~~object; and~~

13   sending the object change set ~~directly~~ from the first system to the second

14 system to cause the second system to apply the object change set to the

15 corresponding object in the second system's cache so as to synchronize the second

16 system with the first system, <u>wherein control is returned to a client coupled to the</u>

17 <u>first system whether or not the second system has completed processing the object</u>

18 <u>change set,</u> wherein the second system is registered in the first system prior to

19 sending the object change information from the first system to the second system,

20 ~~wherein~~ <u>and wherein</u> the registration process causes the first system to know that

21 the second system wants to receive object change information of objects changed

22 in the first system, thereby eliminating the need for the second system to register

23 with every object it is interested in.

1   33. (Currently amended) Electric signals for execution in a computer of a

2 method for providing object change information from a first system to a second

3 system for synchronizing the second system with the first system, the second

4 system having an object cache for storing objects, the method comprising the

5 steps of:

6   changing an object in the first system;

7   determining an object change set which represents changes made to the

8 object in the first system~~; wherein the object change set includes:~~

9   ~~a primary key value that identifies the object; and~~

10   ~~a set of attribute changes which contain the attribute names~~

11   ~~and the new attribute values of attributes that were changed in the~~

12   ~~object; and~~

13   sending the object change set ~~directly~~ from the first system to the second

14 system to cause the second system to apply the object change set to the

15 corresponding object in the second system's cache so as to synchronize the second

16 system with the first system, <u>wherein control is returned to a client coupled to the</u>

17 <u>first system whether or not the second system has completed processing the object</u>

18 <u>change set,</u> wherein the second system is registered in the first system prior to

19 sending the object change information from the first system to the second system,

20 ~~wherein~~ <u>and wherein</u> the registration process causes the first system to know that

21 the second system wants to receive object change information of objects changed

22 in the first system, thereby eliminating the need for the second system to register

23 with every object it is interested in.

1   34. (Currently amended) A computer program product for execution in a

2 computer of a method for providing object change information from a first system

3 to a second system for synchronizing the second system with the first system, the

4 second system having an object cache for storing objects, the computer program

5 product comprising:

6   a module for changing an object in the first system;

7   a module for determining an object change set which represents changes

8 made to the object in the first system~~, wherein an object change set includes:~~

9   ~~a primary key value that identifies the object; and~~

10   ~~a set of attribute changes which contain the attribute names~~

11   ~~and the new attribute values of attributes that were changed in the~~

12   ~~object; and~~

11

13   a module for sending the object change set ~~directly~~ from the first system to

14   the second system to cause the second system to apply the object change

15   information to the corresponding object in the second system's cache so as to

16   synchronize the second system with the first system, <u>wherein control is returned to</u>

17   <u>a client coupled to the first system whether or not the second system has</u>

18   <u>completed processing the object change set,</u> wherein the second system is

19   registered in the first system prior to distributing the object change information

20   from the first system to the second system, ~~wherein~~ <u>and wherein</u> the registration

21   process causes the first system to know that the second system wants to receive

22   object change information of objects changed in the first system, thereby

23   eliminating the need for the second system to register with every object it is

24   interested in.

12